

2D & 3D VISUALIZATIONS

CROSS PLATFORM, TOUCH OPTIMIZED

Christian Liebel, Christian Weyer, Manuel Rauber

thinktecture

thinktecture.com

INTRODUCTION



Christian Liebel

christian.liebel@thinktecture.com

@chris_liebel



Christian Weyer

christian.weyer@thinktecture.com

@christianweyer



Manuel Rauber

manuel.rauber@thinktecture.com

@manuelrauber

TALKING POINTS

2D

- Overview
Browser support, performance
- API
Basics, retina scaling,
animation, interaction
- Examples

3D

- Overview
Basics
- WebGL & three.js
Browser support, controls
- Examples

2D VISUALIZATION

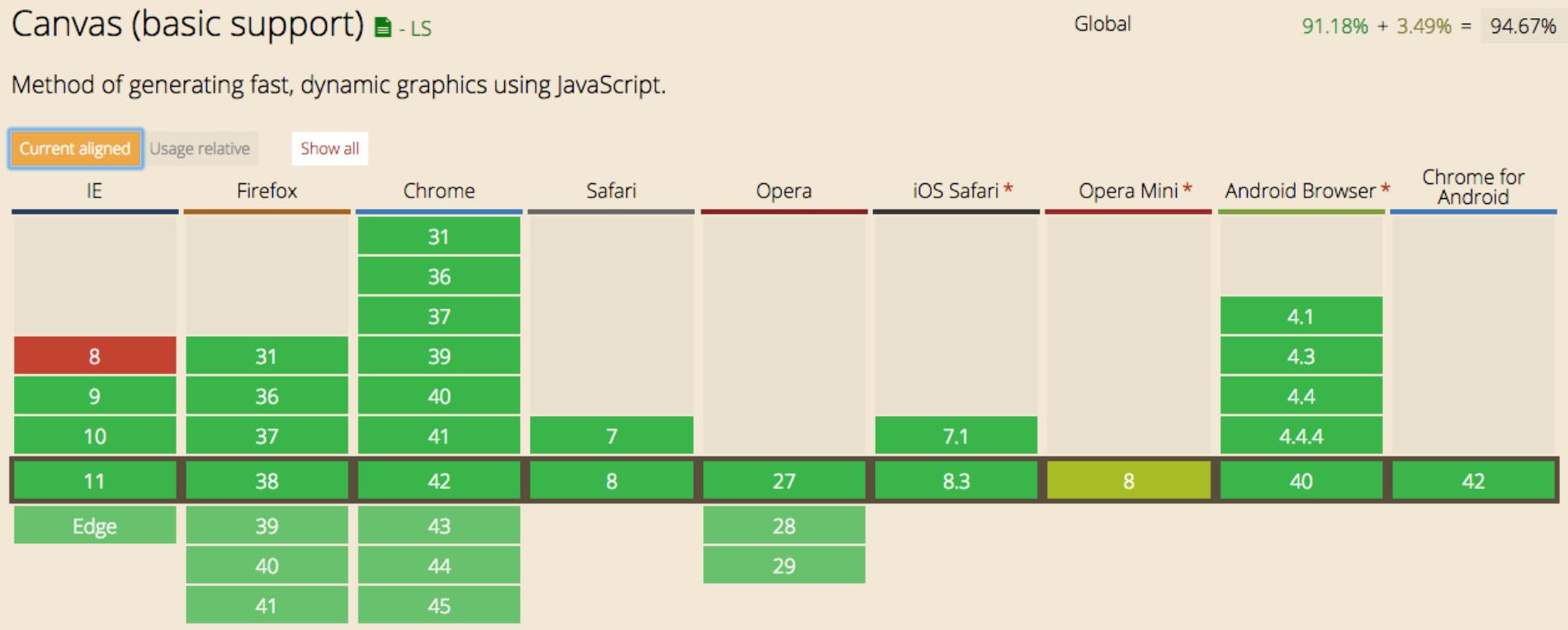
HTML5 CANVAS

Manuel Rauber
@manuelrauber

OVERVIEW

- Plain bitmap for the web
- Cross platform graphic manipulations
- High performance
- Mouse, touch & gamepad support
- Low energy consuming animations

BROWSER SUPPORT (BASIC)

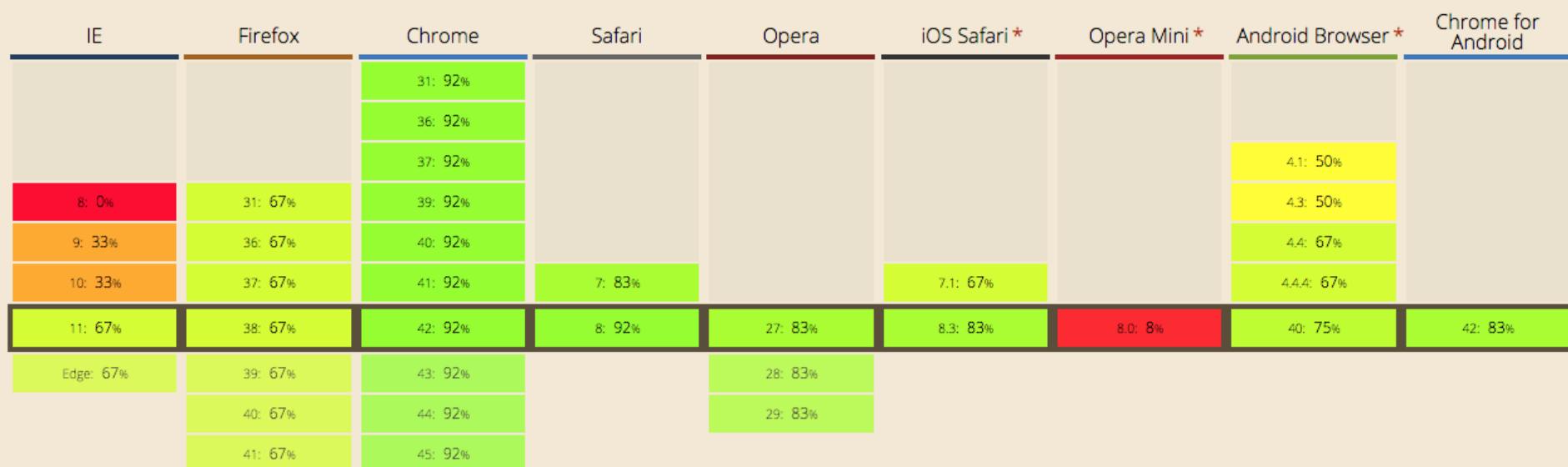


From <http://caniuse.com/#search=canvas>

BROWSER SUPPORT (OVERALL)

Summary

Calculation of support for currently selected criteria



From <http://caniuse.com/#search=canvas>

PERFORMANCE

CHROME @ MACBOOK PRO

Testing in Chrome 42.0.2311.152 on OS X 10.10.3		
	Test	Ops/sec
canvas	<pre>var paper = canvas.getContext('2d'); paper.fillRect(10, 10, 50, 50);</pre>	1,707,273 ±0.98% fastest
svg	<pre>var rect = document.createElementNS("http://www.w3.org/2000/svg", 'rect'); rect.setAttribute("x", 10); rect.setAttribute("y", 10); rect.setAttribute("width", 50); rect.setAttribute("height", 50); svg.appendChild(rect);</pre>	93,927 ±8.99% 95% slower
div	<pre>var div = motherDiv.cloneNode(); div.style.position = "absolute"; div.style.top = "10px"; div.style.left = "10px"; div.style.width = "50px"; div.style.height = "50px"; container.appendChild(div);</pre>	108,994 ±12.54% 94% slower

From <http://jsperf.com/html-vs-svg-vs-canvas/31>

PERFORMANCE

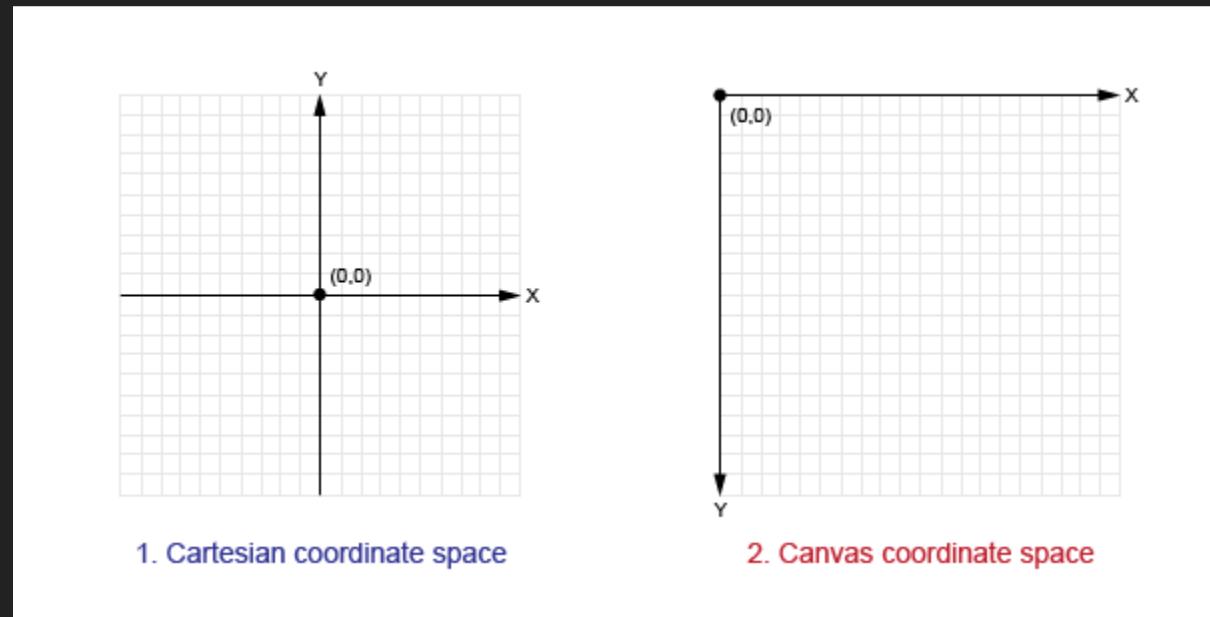
ANDROID @ SAMSUNG GALAXY NOTE

Testing in Android Browser 1.5 (like Chrome 28.0.1500.94) on Samsung (Android 4.4.2)		
	Test	Ops/sec
canvas	<pre>var paper = canvas.getContext('2d'); paper.fillRect(10, 10, 50, 50);</pre>	86,204 ±3.34% fastest
svg	<pre>var rect = document.createElementNS("http://www.w3.org/2000/svg", 'rect'); rect.setAttribute("x", 10); rect.setAttribute("y", 10); rect.setAttribute("width", 50); rect.setAttribute("height", 50); svg.appendChild(rect);</pre>	23,804 ±3.44% 72% slower
div	<pre>var div = motherDiv.cloneNode(); div.style.position = "absolute"; div.style.top = "10px"; div.style.left = "10px"; div.style.width = "50px"; div.style.height = "50px"; container.appendChild(div);</pre>	21,222 ±3.25% 75% slower

COORDINATE SYSTEM

Cartesian: (0/0) bottom left

Canvas: (0/0) top left



From <http://www.sitepoint.com/html5-canvas-tutorial-introduction/>

USAGE

```
<!DOCTYPE html>
<html>
  <head>
    <title>Canvas!</title>
  </head>
  <body>
    <canvas></canvas>
  </body>
</html>
```

FALLBACK

```
<!DOCTYPE html>
<html>
  <head>
    <title>Canvas!</title>
  </head>
  <body>
    <canvas>This text is shown,
    when a browser does not
    support canvas.</canvas>
  </body>
</html>
```

CONTEXTS

2D context: CanvasRenderingContext2D

```
var context = canvas.getContext('2d');
```

WebGL context: WebGLRenderingContext

```
var context = canvas.getContext('webgl');
```

API

DRAWING

```
void fillRect(x, y, width, height);
void strokeRect(x, y, width, height);
void clearRect(x, y, width, height);
void fillText(text, x, y [, maxWidth]);
void strokeText(text, x, y [, maxWidth]);
```

MDN: [CanvasRenderingContext2D](#)

API

PATHS

```
void beginPath();
void closePath();
void moveTo(x, y);
void lineTo(x, y);
void bezierCurveTo(cp1x, cp1y, cp2x, cp2y, x, y);
void rect(x, y, width, height);
void fill();
void stroke();
```

MDN: [CanvasRenderingContext2D](#)

API

PIXEL MANIPULATION

```
ImageData createImageData(width, height);  
ImageData getImageData(sx, sy, sw, sh);  
void putImageData(imagedata, dx, dy);
```

MDN: [CanvasRenderingContext2D](#)

RETINA SUPPORT

●●●○ Telekom.de ⌂

13:56

192.168.0.17

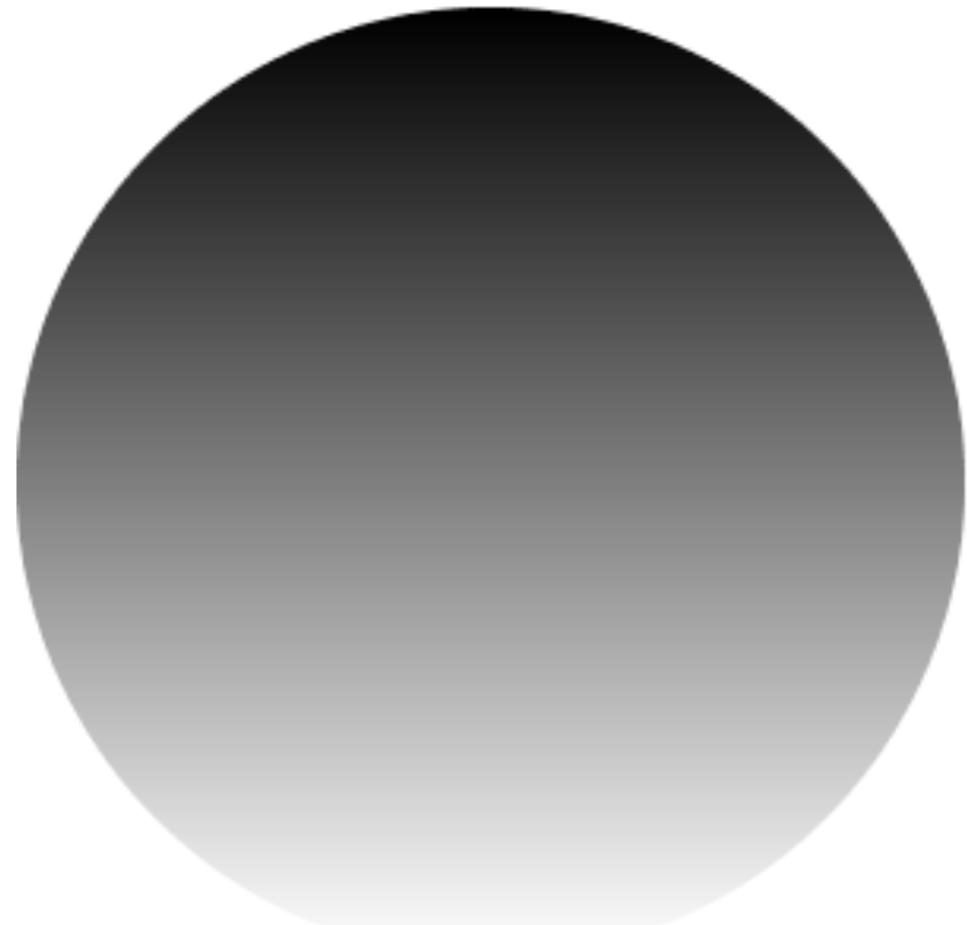
⌚ ⚡ 100% ⌂ ●●●○ Telekom.de ⌂

13:55

192.168.0.17



No Retina



Retina

RETINA - HOW TO

```
function adjustForRetina() {
    if (!window.devicePixelRatio || window.devicePixelRatio <= 1) {
        return;
    }

    var width = canvas.width;
    var height = canvas.height;

    canvas.style.width = width + 'px';
    canvas.style.height = height + 'px';

    canvas.width = width * window.devicePixelRatio;
    canvas.height = height * window.devicePixelRatio;

    context.scale(window.devicePixelRatio, window.devicePixelRatio);
}
```

ANIMATIONS

THE OLD WAY

setTimeout

```
function draw() {  
    // draw routine  
    setTimeout(draw, 1000/60);  
}
```

setInterval

```
setInterval(function () {  
    // draw routine  
, 1000/60);
```

ANIMATIONS

THE MODERN WAY

requestAnimationFrame

```
var animationFrame;  
function startAnimation() {  
    var step = function() {  
        // draw routine  
        animationFrame = window.requestAnimationFrame(step);  
    }  
  
    animationFrame = window.requestAnimationFrame(step);  
}  
  
function stopAnimation() {  
    window.cancelAnimationFrame(animationFrame);  
}
```

MDN: [requestAnimationFrame](#), [cancelAnimationFrame](#)

ANIMATIONS

PROBLEMS WITH "OLD" TECHNIQUES

- If the callback function takes longer than the timer will call it, the callback function gets queued.
- Could lead quickly to an "infinite" amount of queued callback functions.
- Timers continue to work in background tabs, the browser will render invisible animations (wastage of CPU and battery life).
- Especially bad for mobile devices.

From <https://dev.opera.com/articles/better-performance-with-requestanimationframe/>

ANIMATIONS

WHAT REQUESTANIMATIONFRAME DOES

- Only executed when visible to the user (will stop in background tabs!).
- No wastage of CPU and battery life.
- Frame is only drawn when browser is ready and no other frame waits for drawing.
- No "infinite" amount of queued callback functions.

From <https://dev.opera.com/articles/better-performance-with-requestanimationframe/>

ANIMATIONS

WHAT REQUESTANIMATIONFRAME DOESN'T DO

- As the name suggests: Only one frame is requested.
`requestAnimationFrame` has to be called within the callback to get an animation loop.
- You don't know, when the callback will be executed.
- Animations on the same page, where one is in the visible area and one not, will not run synchronously since the hidden one's `requestAnimationFrame` will not be called.

INTERACTION

KEYBOARD & MOUSE SUPPORT

Canvas supports all keyboard & mouse events like `keydown`, `keyup`, `mousedown` or `mousemove`.

```
canvas.addEventListener('mousemove', function (mouseEvent) {  
    // mouse event routine  
});  
  
canvas.addEventListener('keydown', function (keyEvent) {  
    // key down event routine  
});
```

MDN: [MouseEvent](#), [KeyboardEvent](#)

INTERACTION

TOUCH

Canvas supports all touch events like `touchstart`,
`touchmove`, `touchend`.

Consider using a library for recognizing gestures like
[HammerJS](#).

```
canvas.addEventListener('touchmove', function (touchEvent) {  
    // touch move event routine  
});
```

MDN: [Touch events](#), [TouchEvent](#)

EXAMPLES WITHIN GITHUB REPOSITORY

1. Basic circle
2. Circle with gradient
3. Retina support
4. requestAnimationFrame
5. Time-based animations
6. Mouse Move
7. Touch Move
8. Manipulating a rectangle with HammerJS

EXAMPLES WITHIN THE WEB

- Gesture recognition: <http://revealjs.herokuapp.com/>
- Video explosion:
<http://www.craftymind.com/factory/html5video/CanvasVideoExplosion.html>
- Zen Photon Garden: <http://zenphoton.com/>
- Zoom Charts: <https://zoomcharts.com/developers/en/net-chart/examples/items/link-items.html>
- Many more examples @<http://www.canvasdemos.com/>

VIDEO EXPLOSION

From <http://www.craftymind.com/factory/html5video/CanvasVideo.html>



ZEN PHOTON GARDEN

From <http://zenphoton.com/>

ZOOM CHARTS

From <https://zoomcharts.com/developers/en/net-chart/examples/items/link-items.html>

LIVE DEMO

3D VISUALIZATION

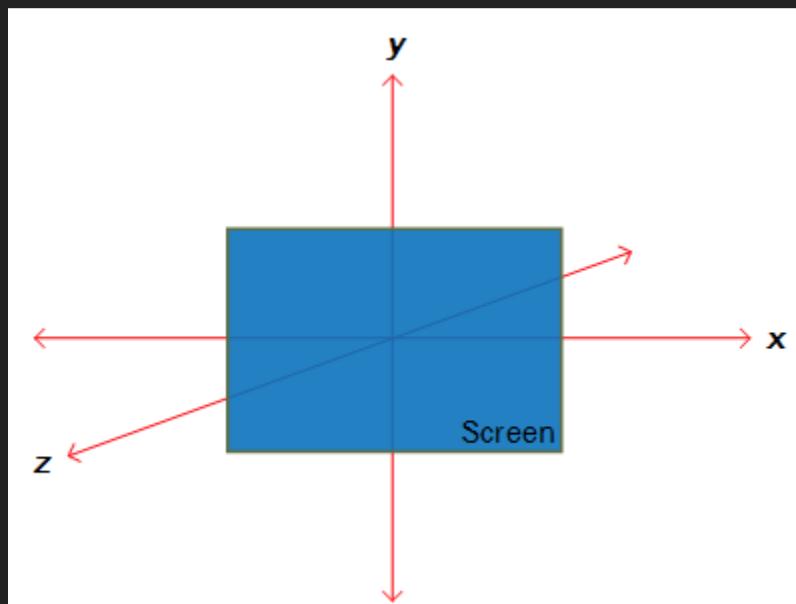
Christian Liebel
@chris_liebel

3D CONTENT IN A 3D WORLD WITH 2D SCREENS

scenes, cameras, rendering

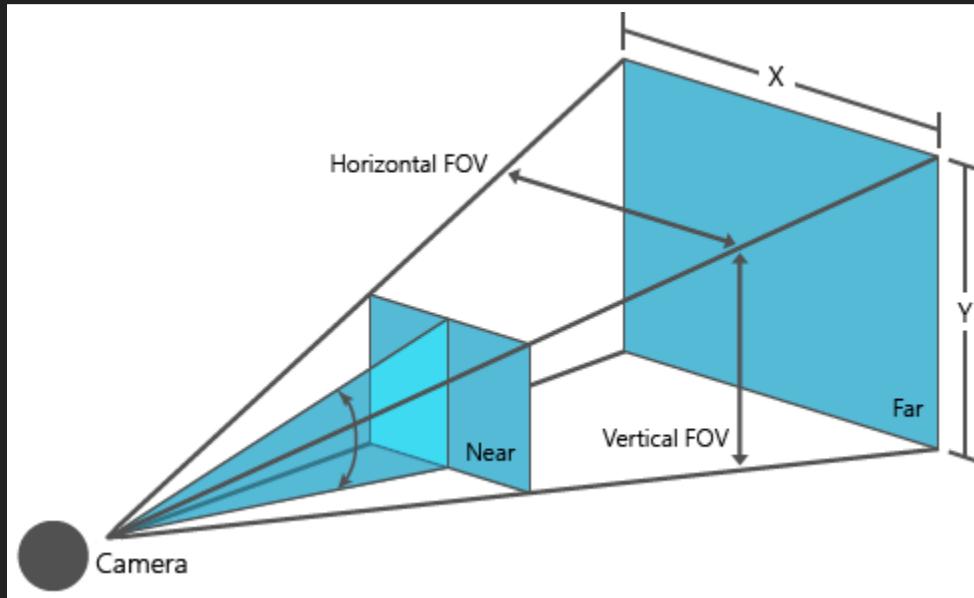
COORDINATE SYSTEM

Right-handed



From [https://msdn.microsoft.com/library/dn479430\(v=vs.85\).aspx](https://msdn.microsoft.com/library/dn479430(v=vs.85).aspx)

CAMERA FRUSTUM



From [https://msdn.microsoft.com/library/dn479430\(v=vs.85\).aspx](https://msdn.microsoft.com/library/dn479430(v=vs.85).aspx)

WEBGL

- Web Graphics Library
- Hardware accelerated (GPU)
- New canvas rendering context `webgl`

PROVEN TECHNOLOGY

- Based on OpenGL for Embedded Systems
- Blender & Unity can export to WebGL
- ANGLE (Chrome/Firefox): Rendering on Direct3D

THREE.JS

- Native WebGL *is* complex
- Open-source JavaScript library
- High-level API abstracting WebGL (and more)

STRUCTURE

- Geometry: shape (cube, sphere, ...)
- Material: color, texture, light reflection
- Mesh: 3D object (geometry + material)

BROWSER SUPPORT



From <http://caniuse.com/#search=webgl>

MOBILE PLATFORM SUPPORT

- iOS 8+
- Android
 - WebView 36+
 - Crosswalk
- Windows (Phone) 8.1+

CONTROLS

- Mouse/Keyboard
- Touch
- Gamepad

A WORD ON TOUCH EVENTS...

- Pointer Events vs. Touch Events
- Browser support varies on desktop
- three.js controls only support Touch Events

A WORD ON GAMEPADS...

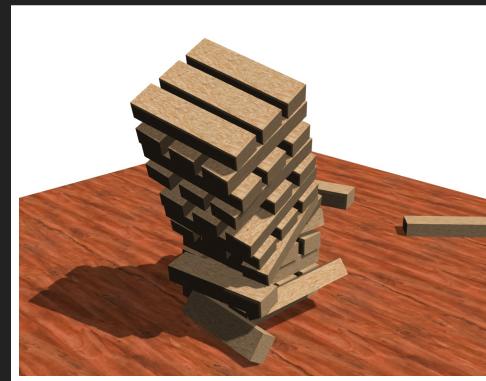
- HTML5 Gamepad API
- Polling (`requestAnimationFrame`)
- Supported by Chrome, Firefox, MS Edge, Opera

THE CREATION OF THE WORLD

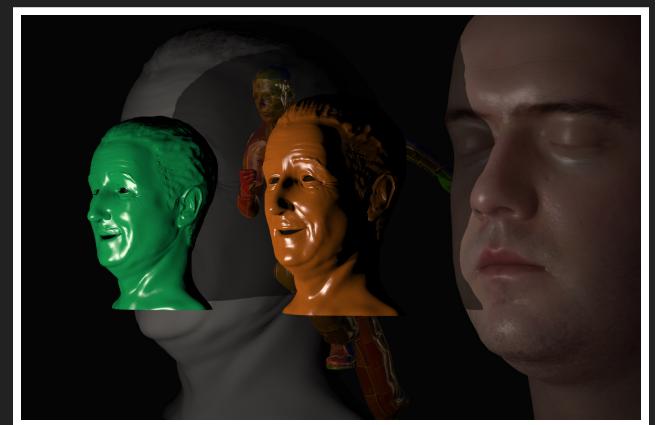
EVEN MORE SAMPLES...



Flight Arcade



Physijs



three.js samples

SUMMARY

- Cross platform high-performance graphic manipulation
- 2D & 3D
- Hardware acceleration
- Supported by all major browsers
- Touch-enabled, gamepad-enabled

THANK YOU!

GitHub repository:

<https://github.com/thinktecture/DotNetUserGroupKA2015>

thinktecture

thinktecture.com

RESOURCES

- GitHub repository: <https://github.com/thinktecture/DotNetUserGroupKA2015>
- Canvas browser support: <http://caniuse.com/#search=canvas>
- Cordova: <https://cordova.apache.org/>
- Ionic: <http://ionicframework.com/>
- Gulp: <http://gulpjs.com>
- TypeScript: <http://www.typescriptlang.org/>
- CanvasRenderingContext2D:
<https://developer.mozilla.org/en/docs/Web/API/CanvasRenderingContext2D>
- WebGLRenderingContext:
<https://developer.mozilla.org/en/docs/Web/API/WebGLRenderingContext>
- requestAnimationFrame: <https://developer.mozilla.org/en-US/docs/Web/API/window/requestAnimationFrame>